1. A computer system comprising:

a hierarchical memory system, including a first local storage, and a main storage; and

a first memory access unit coupled to the hierarchical memory system capable of processing a plurality of memory access instructions that includes

(a) a first instruction that specifies a first address and a first value, wherein processing the first instruction by the first memory access unit causes the first value to be stored at a location in the first local storage that is associated with the first address, and

(b) a second instruction that specifies the first address, wherein processing of the second instruction by the first memory access unit after processing the first instruction is such that the first memory access unit complete processing of the second instruction after the first value is stored at a location in the main storage that is associated with the first address.

2. The computer system of claim 1 wherein the plurality of memory access instructions further includes

(c) a third instruction that specifies the first address, wherein processing of the third instruction by the first memory access unit causes a value to be retrieved from a location in the first local storage that is associated with the first address, and

(d) a fourth instruction that specifies the first address, wherein processing of the fourth instruction by the first memory access unit prior to processing the third instruction causes the value retrieved during processing the third instruction to be a value that was retrieved from a location in the main storage that is associated with the first address at a time after the fourth instruction was begun to be processed.

3. The computer system of claim 1 wherein the hierarchical memory system further includes a second local storage, and the computer system further comprises a second memory access unit coupled to the hierarchical memory system capable of processing the plurality of memory access, and the plurality of instructions further includes:

(c) a third instruction that specifies the first address, wherein processing of the third instruction by the second memory access unit causes a value to be retrieved from a location in the second local storage that is associated with the first address, and

(d) a fourth instruction that specifies the first address, wherein processing of the fourth instruction by the second memory access unit prior to processing the third instruction and after the first memory access unit has completed processing the second instruction causes the value retrieved during processing the third instruction to be a value that was retrieved from a location in the main storage that is associated with the first address at a time after the fourth instruction was begun to be processed, whereby the value caused to be retrieved by the processing of the third instruction by the second memory access unit is the first value, which was specified in the first instruction which was processed by the first memory access unit.

4. A computer processor for use in a multiple processor system in which the computer processor is coupled to one or more other processors through a memory system, the computer processor comprises a memory access unit configured to access the memory system by processing a plurality of memory access instructions, including

(a) a first instruction that specifies a first address and a first value, wherein processing the first instruction causes the first value to be stored at a location in the memory system that is associated with the first address, such that for at least some period of time the one or more other processors do not have access to the first value, and

(b) a second instruction that specifies the first address, wherein processing of the second instruction after processing the first instruction is such that the processing of the second instruction completes after the first value is accessible to each of the one or more other processors.

5. The computer processor of claim 4 wherein the plurality of memory access instructions further includes

(c) a third instruction that specifies a second address, wherein processing of the third instruction causes a value to be retrieved from a location in the memory system that is associated with the second address, and

(d) a fourth instruction that specifies the second address, wherein processing of the fourth instruction prior to processing the third instruction causes the third instruction to retrieve a value that was previously stored in the memory system by one of the one or more other processors.

6. A multiple processor computer configured to use a storage system, the computer comprising a plurality of memory access units, including:

a first memory access unit responsive to execution of instructions by a first instruction processor, wherein the first memory access unit is coupled to the storage system; and

a second memory access unit responsive to execution of instructions by a second instruction processor, wherein the second memory access is coupled to the storage system;

wherein the first and the second memory access units are each capable of issuing memory access messages to the storage system and receiving return messages from the storage system in response to the memory access messages, the memory access messages and return messages including:

a first memory access message that specifies a first address and a first value for causing the first value to be

stored at a first location in storage system that is associated with the first address;

a first return message that is a response to the first memory access message, indicating that the first value has been stored in the storage system at a location that is associated with the first address and that is accessible to the memory access unit receiving the first return message;

a second return message indicating that the first value has been stored in the storage system at a location that is associated with the first address and that is accessible to each of the plurality of memory access units.

7. The multiple processor computer of claim 6 wherein the memory access messages and return messages further include a second memory access message that specifies the first address, and wherein the second return message is a response to the second memory access message.

8. The multiple processor computer of claim 7 wherein the first memory access unit is configured to issue the first memory access message in response to execution of a first processor instruction that specifies the first address and the first value, and is configured to issue the second memory access message in response to execution of a second processor instruction that specifies the first address.

9. A memory system for use in a multiple processor computer system in which the memory system is coupled to a plurality of computer processors, wherein the memory system comprises a plurality of local storages, including a first local storage unit and other local storage units, and each local storage unit is capable of processing a plurality messages received from a corresponding one of the computer processors, the plurality of messages includes:

(a) a first message that specifies a first address and a first value, wherein processing the first message by the first local storage unit causes the first value to be stored at a location in the local storage unit that is associated with the first address, such that, for at least a period of time, the other local storage units do not have access to the first value, and

(b) a second message that specifies the first address, wherein processing of the second message by the first local storage unit after processing the first message is such that the processing of the second message completes after the first value can be accessed by each of the other local storage units.

10. The memory system of claim 9 wherein the plurality of memory access messages further includes

(c) a third message that specifies a second address, wherein processing of the third message causes a value to be retrieved from a location in the first local storage that is associated with the second address and to be sent to the corresponding computer processor, and

(d) a fourth message that specifies the second address, wherein processing of the fourth message prior to processing the third message guarantees that the value caused to be sent in processing the third message is a value that was previously stored in the memory system by one of the other processors.

11. The memory system of claim 9 further comprising:

a main storage wherein values stored in the main storage are accessible to each of the plurality of local storages; and

a controller configured to transfer data between the main storage and the plurality of local storages according to a plurality of stored rules.

12. The memory system of claim 11 wherein the plurality of stored rules includes:

a rule for initiating a transfer of the first value from the local storages to the main storage after processing the first message and prior to processing the second message.

13. A computer processor for use in a multiple processor computer system in which the computer processor and one or more other computer processors are coupled to a storage system, the computer processor comprising:

a storage capable of holding a sequence of instructions, wherein the sequence of instructions includes a first instruction that specifies a first address range and a second address range, and includes a first set of instructions that each specifies an address in the first address range and that are prior to the first instruction in the sequence, and a second set of instructions that each specifies an address in the second address range and that are after the first instruction in the sequence;

an instruction scheduler coupled to said storage, wherein the instruction scheduler is configured to issue instructions in the sequence of instructions such that instructions in the second set of instructions do not issue prior to all of the instructions in the first set of instructions completing.

14. The computer processor of claim 13 wherein the first set of instructions includes instructions that may result in data previously stored in the storage system by one of the one or more other processors at an address in the first address range being transferred to the computer processor.

15. The computer processor of claim 14 wherein the second set of instructions includes instructions that each initiates a transfer of data from the computer processor to for storage at an address in the second address range such that the data is accessible to the one or more other processors.

16. The computer processor of claim 14 wherein the second set of instructions includes instructions that may result in data previously stored in the storage system by one of the one or more other processors at an address in the second address range being transferred to the computer processor.

17. The computer processor of claim 13 wherein the first set of instructions includes instructions that each completes after the instruction schedule receives a corresponding notification from the storage system that a value has been stored in the storage system at an address in the first address range such that the value is accessible to the one or more other processors.

18. The computer processor of claim 17 wherein the second set of instructions includes instructions that initiate a transfer of data from the computer processor to for storage at an address in the second address range such that the data is accessible to the one or more other processors.

19. The computer processor of claim 17 wherein the second set of instructions includes instructions that may result in data previously stored in the storage system by one of the one or more other processors at an address in the second address range being transferred to the computer processor.

20. A method for accessing a memory system from a processor in a multiple processor computer system, comprising:

(a) in a first processor that is coupled to a first local storage in the memory system, processing a first instruction that specifies a first address and a first value, including storing the first value at a location in the first local storage that is associated with the first address, and

(b) in the first processor, after processing the first instruction, processing a second instruction that specifies the first address, wherein processing of the second instruction completes after the first value is stored at a location in a shared storage in the memory system that is associated with the first address.

21. The method of claim 20 further comprising:

(c) in a second processor that is coupled to a second local storage in the memory system, processing a third instruction that specifies the first address, including retrieving a value from a location in the second local storage that is associated with the first address, and

(d) in the second processor, processing a fourth instruction that specifies the first address prior to processing the third instruction and after the first processor has completed processing the second instruction, including retrieving the first value from the location in the shared storage that is associated with the first address and storing the first value at a location in the second local storage that is associated with the first address, whereby the value retrieved in the processing of the third instruction is the first value, which was specified in the first instruction.

22. A method for providing data storage for a plurality of computer processors in a memory system that includes a plurality of local storages, including a first local storage unit and other local storage units, the method comprising:

receiving at the first local storage a first message from a corresponding one of the plurality of computer processors, wherein the first message specifies a first address and a first value;

processing the first message by the first local storage unit including storing the first value at a location in the local storage unit that is associated with the first address, such that, for at least a period of time, the other local storage units do not have access to the first value;

receiving at the first local storage a second message from the corresponding one of the plurality of computer processors, wherein the second message specifies the first address;

processing the second message by the first local storage unit after processing the first message such that the processing of the second message completes after the first value can be accessed by each of the other local storage units.

23. The method of claim 22 further comprising:

receiving by the first local storage unit a third message from the corresponding one of the plurality of computer processors, wherein the third message specifies a second address;

processing the third message including retrieving a value from a location in the first local storage that is associated with the second address and sending the retrieved value to the corresponding one of the plurality of computer processors;

receiving by the first local storage unit a fourth message from the corresponding one of the plurality of computer processors, wherein the fourth message specifies the second address; and

processing of the fourth message prior to processing the third message;

wherein the value sent in processing the third message is a value that was previously stored in the memory system by one of the other processors.

24. The method of claim 22 wherein the memory system includes a main storage wherein values stored in the main storage are accessible to each of the plurality of local storages, and the method further comprises:

accessing a plurality of stored rules; and

transferring data between the main storage and the plurality of local storages according to the accessed rules.

25. The method of claim 24 wherein transferring data between the main storage and the plurality of local storages includes initiating a transfer of the first value from the local storages to the main storage after processing the first message and prior to processing the second message.

26. A method for scheduling instructions in a computer processor, comprising:

accepting a sequence of instructions that includes a first instruction that specifies a first address range and a second address range, a first set of instructions that each specifies an address in the first address range and that are prior to the first instruction in the sequence, and a second set of instructions that each specifies an address in the second address range and that are after the first instruction in the sequence;

executing the first instruction, including waiting for all instructions in the first set to complete; and

executing instructions in the second set only after executing the first instruction.